

REMARKS

Applicants appreciate the Examiner's thorough examination of the present application as evidenced by the Office Action of January 21, 2003 (hereinafter "Office Action"). In response, Applicants have amended independent Claims 1, 9, 20, 28, 39, and 47 to clarify that the format argument "is a pointer to a memory location in an address space of the application." Applicants have amended various other claims to be consistent with the amendments made to the independent claims. In addition, Claims 19, 38, and 57 have been amended to correct their dependencies. Applicants respectfully submit that the cited references, either alone or in combination, fail to disclose or suggest, all of the recitations of independent Claims 1, 9, 20, 28, 39, and 47. Therefore, Applicants respectfully submit that all pending claims are in condition for allowance. Favorable reconsideration of all pending claims is respectfully requested for at least the reasons discussed hereafter.

Dependent Claims 19, 38, and 57 Satisfy the Requirements of 35 U.S.C. §112

Dependent Claims 19, 38, and 57 stand rejected under 35 U.S.C. §112, ¶2 as being indefinite. In particular, these claims incorrectly depend from Claims 17, 36, and 55, respectively. In response, Applicants have amended Claims 19, 38, and 57 to depend from Claims 18, 37, and 56, respectively. Applicants, therefore, respectfully submit that Claims 19, 38, and 57 satisfy the requirements of 35 U.S.C. §112.

Independent Claims 1, 9, 20, 28, 39, and 47 are Patentable

Independent Claims 1, 9, 20, 28, 39, and 47 stand rejected under 35 U.S.C. §102(e) as being anticipated by U. S. Patent No. 5,983,366 to King (hereinafter "King"). Independent Claims 1, 9, 20, 28, 39, and 47 are directed to methods, systems, and computer program products for printing data from an application in which a print function is invoked with a format argument and the format argument is saved in a deferred trace data buffer. These independent claims have been amended, however, to clarify that the format argument "is a pointer to a memory location in an address space of the application." Embodiments including this aspect of the present invention are described, for example, at page 11, lines 5 - 16 of the Specification.

In rejecting Claim 10, the Office Action acknowledges that King does not disclose "the format argument being a pointer to a memory location in an address space of the application, and saving the pointer in the deferred trace data buffer." (Office Action, page 10). The Office Action does assert, however, that U. S. Patent No. 6,282,701 to Wygodny et al. (hereinafter "Wygodny") teaches "displaying a pointer (for example, variable names) and the contents of the memory referred to by the pointer as part of a trace output display..." (Office Action, page 11). In rejecting Claim 15, the Office Action again acknowledges that King does not disclose "the format argument being a pointer to a memory location in an address space of the application..." (Office Action, page 7). The Office Action does assert, however, that the document "The Visual C++ Debugging Environment" authored by Keith Bugg (hereinafter "Bugg") teaches "a format argument a debugging information output command being a pointer to a memory location in an address space of an application..." (Office Action, page 8).

To establish a prima facie case of obviousness, the prior art reference or references when combined must teach or suggest *all* the recitations of the claims, and there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. §2143. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. M.P.E.P. §2143.01, citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990). As recently emphasized by the Court of Appeals for the Federal Circuit, to support combining references, evidence of a suggestion, teaching, or motivation to combine must be **clear and particular**, and this requirement for clear and particular evidence is not met by broad and conclusory statements about the teachings of references. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). In an even more recent decision, the Court of Appeals for the Federal Circuit has stated that, to support combining or modifying references, there must be **particular** evidence from the prior art as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000).

Applicants respectfully submit that the King, Wygodny, and Bugg references contain no description therein to suggest to or motivate one skilled in the art to modify King's computer program tracing system with the teachings of either Wygodny or Bugg. In fact, Applicants respectfully submit that the disclosures of King, Wygodny, and Bugg teach against such a combination as the resulting computer program tracing system would be inoperable.

King explains at column 19, lines 17 - 27, that a computer program may perform a trace by calling a trace macro, which in turns calls a trace function that includes a numerical identification of the trace message and a pair of parameters. In King's example, the trace message is identified as number 7292. The parameters and trace identification are packed into a message and transmitted from the data processing system 252 to the host processor 254. (King, col. 19, lines 40 - 44). Once the message is received at the host processor 254, the host processor 254 looks "up the trace id and then correctly unpack the trace message and display the trace string and its parameters in the way defined in the trace control file..." (King, col. 19, lines 50 - 52). Because King describes processing the trace message, which includes a trace identification and parameters, on a different processor (host processor 254) than the processor executing the computer program that is being traced (data processing system 252), replacing the trace identification with a pointer would not work because the host processor 254 does not have access to the address space of the data processing system 252. That is, a pointer to a memory location in the data processing system 252 is useless to the host processor 254.

Accordingly, Applicants respectfully submit that one skilled in the art would not be motivated to replace the trace identification described in King with a pointer as described in Wygodny or Bugg as such a replacement would render King's computer program tracing system inoperable.

For at least the foregoing reasons, Applicants respectfully submit that independent Claims 1, 9, 20, 28, 39, and 47 are patentable over the cited references and that dependent Claims 2 - 8, 10 - 19, 21 - 27, 29 - 38, 40 - 46, and 48 - 57 are patentable at least by virtue of their depending from an allowable claim.

In re: Fluke et al.
Serial No.: 09/607,074
Filed: June 29, 2000
Page 10 of 16

Dependent Claims 7, 26, and 45 are Separately Patentable

With regard to Claims 7, 26, and 45, these claims include all of the recitations from independent Claims 1, 20, and 39, respectively, and are, therefore, patentable over the cited references for at least the reasons stated above. In addition, Applicants submit that these claims are separately patentable as none of the cited references described or suggest "saving...a memory contents comprising the address space of the application in a non-volatile medium."

The Office Action alleges that Bugg teaches "sending debugging output, including format and data arguments to a file, a debugger, and/or a message window..." (Office Action, page 7). Applicants respectfully submit that Bugg describes the ability to send a processed debug report, including a file name, linenumber, and a formatted message, to a file, debugger, or message window. Applicants submit that Bugg contains no description of saving the address space of the application in which the `_CrtDbgReport()` function is called to a non-volatile medium.

Accordingly, Applicants respectfully submit that Claims 7, 26, and 45 are separately patentable for at least these additional reasons.

CONCLUSION

In light of the above amendments and remarks, Applicants respectfully submit that the above-entitled application is now in condition for allowance. Favorable reconsideration of this application, as amended, is respectfully requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (919) 854-1400.

In re: Fluke et al.
Serial No.: 09/607,074
Filed: June 29, 2000
Page 11 of 16

It is not believed that an extension of time and/or additional fee(s), including fees for net addition of claims, are required, beyond those that may otherwise be provided for in documents accompanying this paper. In the event, however, that an extension of time is necessary to allow consideration of this paper, such an extension is hereby petitioned under 37 C.F.R. §1.136(a). Any additional fees believed to be due in connection with this paper may be charged to IBM's Deposit Account No. 50-0563.

Respectfully submitted,



D. Scott Moore
Registration No. 42,011

Customer Number:



20792

PATENT TRADEMARK OFFICE

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Box Non-Fee Amendment, Commissioner for Patents, Washington, DC 20231, on April 21, 2003.



Traci A. Brown

Date of Signature: April 21, 2003

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Claims:

Please amend the following Claims by adding the language that is underlined ("___") and by deleting the language that is enclosed within brackets ("[]"):

1. (Amended) A method of printing data from an application, comprising the steps of:

invoking a print function with a format argument that is a pointer to a memory location in an address space of the application and at least one data argument from the application;

saving the format argument and the at least one data argument in a deferred trace data buffer;

returning to the application that invoked the print function; then

processing the deferred trace data buffer to print the at least one data argument.

7. (Amended) A method as recited in Claim 1, further comprising the step of:
saving the deferred trace data buffer and a memory contents comprising [an] the address space of the application in a non-volatile storage medium.

9. (Amended) A method of printing data from an application, comprising the steps of:

invoking a print function with a format argument that is a pointer to a memory location in an address space of the application from the application;

saving the format argument in a deferred trace data buffer;

returning to the application that invoked the print function; then

processing the deferred trace data buffer to print the format argument.

10. (Amended) A method as recited in Claim 9, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the step of saving the format argument in the deferred trace data buffer comprises the step of:

saving the pointer in the deferred trace data buffer

15. (Amended) A method as recited in Claim 9, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the step of saving the format argument in the deferred trace data buffer comprises the step of:

saving a contents of the memory location in the address space of the application that is referenced by the pointer in the deferred trace data buffer.

19. (Amended) A method as recited in Claim [17] 18, wherein the step of saving the contents of the memory location in the address space that is referenced by the pointer in the deferred trace data buffer is performed on a first computing machine and the step of processing the deferred trace data buffer to print the contents of the memory location in the address space of the application that is referenced by the pointer is performed on a second computing machine, the second computing machine being different from the first computing machine and having access to the deferred trace data buffer via the non-volatile storage medium.

20. (Amended) A system for printing data from an application, comprising:
means for invoking a print function with a format argument that is a pointer to a memory location in an address space of the application and at least one data argument from the application;

means for saving the format argument and the at least one data argument in a deferred trace data buffer;

means for returning to the application that invoked the print function; and

means for processing the deferred trace data buffer to print the at least one data argument after returning to the application that invoked the print function.

26. (Amended) A system as recited in Claim 20, further comprising:

means for saving the deferred trace data buffer and a memory contents comprising [an] the address space of the application in a non-volatile storage medium.

28. (Amended) A system for printing data from an application, comprising:
means for invoking a print function with a format argument that is a pointer to a memory location in an address space of the application from the application;
means for saving the format argument in a deferred trace data buffer;
means for returning to the application that invoked the print function; and
means for processing the deferred trace data buffer to print the format argument after returning to the application that invoked the print function.

29. (Amended) A system as recited in Claim 28, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the means for saving the format argument in the deferred trace data buffer comprises:
means for saving the pointer in the deferred trace data buffer.

34. (Amended) A system as recited in Claim 28, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the means for saving the format argument in the deferred trace data buffer comprises:
means for saving a contents of the memory location in the address space of the application that is referenced by the pointer in the deferred trace data buffer.

38. (Amended) A system as recited in Claim [36] 37, wherein the means for saving the contents of the memory location in the address space that is referenced by the pointer in the deferred trace data buffer executes on a first computing machine and the means for processing the deferred trace data buffer to print the contents of the memory location in the address space of the application that is referenced by the pointer executes on a second computing machine, the second computing machine being different from the first computing machine and having access to the deferred trace data buffer via the non-volatile storage medium.

39. (Amended) A computer program product for printing data from an application,

comprising:

a computer readable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for invoking a print function with a format argument that is a pointer to a memory location in an address space of the application and at least one data argument from the application;

computer readable program code for saving the format argument and the at least one data argument in a deferred trace data buffer;

computer readable program code for returning to the application that invoked the print function; and

computer readable program code for processing the deferred trace data buffer to print the at least one data argument after returning to the application that invoked the print function.

45. (Amended) A computer program product as recited in Claim 39, further comprising:

computer readable program code for saving the deferred trace data buffer and a memory contents comprising [an] the address space of the application in a non-volatile storage medium.

47. (Amended) A computer program product for printing data from an application, comprising:

a computer readable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for invoking a print function with a format argument that is a pointer to a memory location in an address space of the application from the application;

computer readable program code for saving the format argument in a deferred trace data buffer;

computer readable program code for returning to the application that invoked

the print function; and

computer readable program code for processing the deferred trace data buffer to print the format argument after returning to the application that invoked the print function.

48. (Amended) A computer program product as recited in Claim 47, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the computer readable program code for saving the format argument in the deferred trace data buffer comprises:

computer readable program code for saving the pointer in the deferred trace data buffer.

53. (Amended) A computer program product as recited in Claim 47, [wherein the format argument is a pointer to a memory location in an address space of the application, and] wherein the computer readable program code for saving the format argument in the deferred trace data buffer comprises:

computer readable program code for saving a contents of the memory location in the address space of the application that is referenced by the pointer in the deferred trace data buffer.

57. (Amended) A computer program product as recited in Claim [55] 56, wherein the computer readable program code for saving the contents of the memory location in the address space that is referenced by the pointer in the deferred trace data buffer executes on a first computing machine and the computer readable program code for processing the deferred trace data buffer to print the contents of the memory location in the address space of the application that is referenced by the pointer executes on a second computing machine, the second computing machine being different from the first computing machine and having access to the deferred trace data buffer via the non-volatile storage medium.